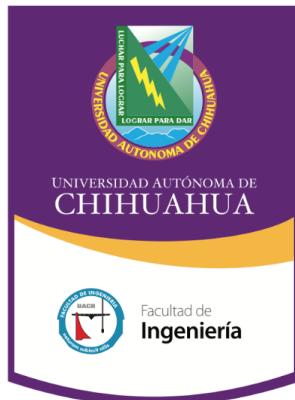


UNIVERSIDAD AUTÓNOMA DE CHIHUAHUA
FACULTAD DE INGENIERÍA



CIRCUITOS LÓGICOS II
GRUPO: 7HW1

PRÁCTICA NO. 5
“SUMADOR DE UN BIT”

EQUIPO

“sudo rm -rf / --no-preserve-root”

INTEGRANTES DEL EQUIPO

JESÚS ALEJANDRO JIMÉNEZ HERNÁNDEZ
OSCAR ALEJANDRO SERRANO PIZARRO
EMILIO ALBERTO SALAS RIOS

FECHA DE ENTREGA

DOMINGO 2 DE OCTUBRE DE 2022

DOCENTE

JESÚS MANUEL MUÑOZ LARGUERO

ÍNDICE

INTRODUCCIÓN	3
Objetivos	3
DESARROLLO	4
1. Configurar el Software Quartus II para que pueda funcionar con el CPLD MAX II.	4
2. Armar un circuito de salida que incluya las entradas y salidas necesarias para operar un circuito decodificador 7447 o 7448.	4
3. Compilar y grabar en el CPLD un programa que mediante un diagrama esquemático y código VHDL un código para nuestro decodificador.	5
CONCLUSIÓN	9
BIBLIOGRAFÍA	10

INTRODUCCIÓN

En esta práctica se nos pide configurar el CPLD para que realice la labor de sumador, es decir, este sumará los valores de dos bits de entrada (A y B) y un carry de entrada y nos mostrará dichos valores a través de dos diodos LED que representaran la suma y el carry de salida.

Objetivos

- Comprender el funcionamiento y simulación de un sumador binario de un bit.
- Configurar un sumador binario de un bit completo utilizando lenguaje VHDL y con diagrama de circuitos lógicos.
- Armar el circuito y probarlo.

DESARROLLO

- Lo primero que realizamos fue la configuración del software Quartus II para poder utilizar un circuito integrado EMP 240 T 100 C 5 de la familia de CPLDs MAX II. La herramienta seleccionada para la simulación fue ModelSim-Altera, utilizando código VHDL.
- En el ambiente de trabajo del software, creamos un nuevo archivo VHDL donde escribimos el código. En este paso nos aseguramos de que el nombre del módulo fuera igual al nombre de nuestro proyecto.
- Realizamos el análisis y elaboración del código y guardamos los cambios del archivo para comprobar que no existieran errores.
- Una vez que no detectó errores compilamos el programa.
- Seguidamente hicimos la configuración de los pines de entrada y de salida.
- En la entrada tenemos:
 - Pin 36 → Entrada a sumar 'A' → Botón A
 - Pin 38 → Entrada a sumar 'B' → Botón B
 - Pin 40 → Entrada del carry 'C' → Botón C
- Las compuertas de salida se encuentran de la siguiente forma:
 - Pin 76 → Salida resultado de la suma → LED 'SUM'
 - Pin 74 → Salida de carry → LED 'Cout'
- Después conectamos el USB Blaster a la tablilla CPLD y alimentamos la misma con un voltaje de 5 V.
- Una vez que el software reconoce el USB Blaster procedimos a subir el código compilado a la tablilla CPLD, seleccionando todas las opciones de "Program/Configure" que aparecieron en el programador del software.
- Por último, guardamos los cambios del archivo y comprobamos que funcionará físicamente en el protoboard. El circuito debe mostrar en los LEDs de salida el resultado correspondiente a la suma de las entradas de acuerdo con la siguiente tabla de verdad.

ENTRADA

SALIDA

A	B	C	Cout	SUM
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1

0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

- Para la segunda parte de esta práctica configuramos los pines de entrada y salida exactamente igual, con la diferencia que, en lugar de escribir el código, la configuración del circuito la realizamos dibujando el diagrama de circuitos lógicos. El resultado fue el mismo.

Evidencias

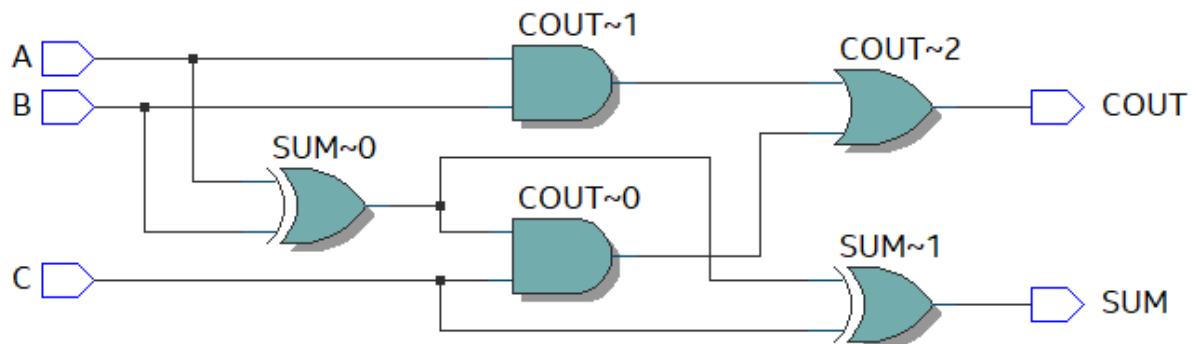
- Código utilizado durante la práctica

```

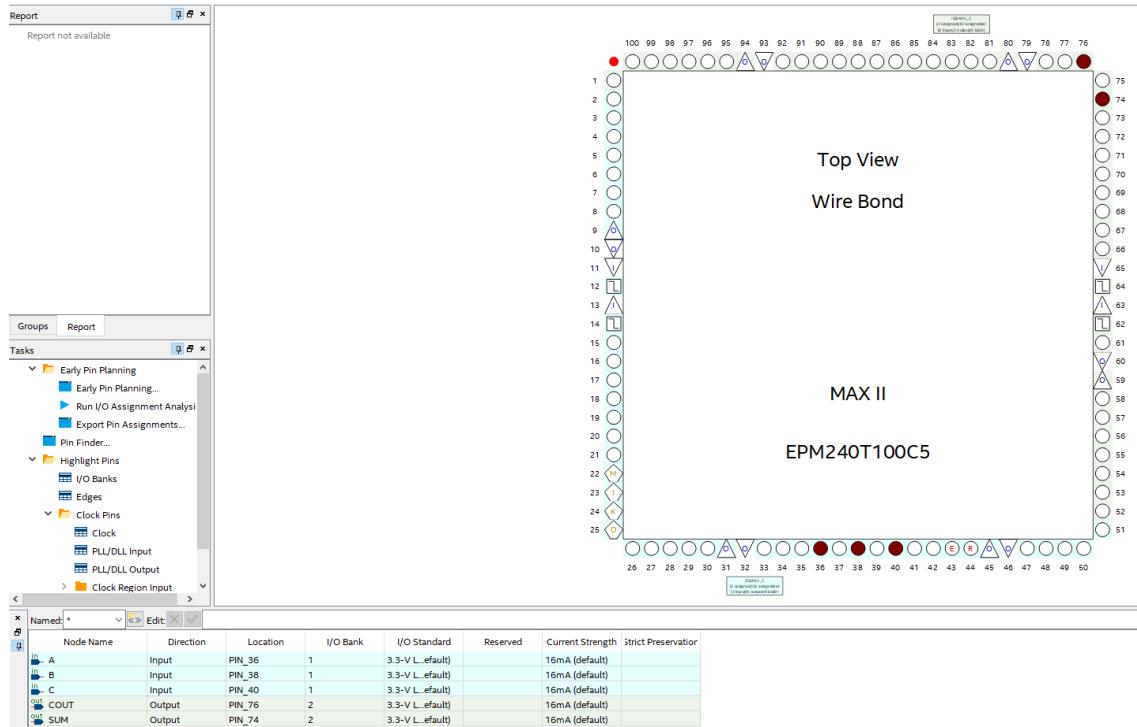
1  library IEEE;
2  use IEEE.std_logic_1164.all;
3
4  entity Practica_5 is
5    Port (
6      A: in STD_LOGIC;
7      B: in STD_LOGIC;
8      C: in STD_LOGIC;
9      SUM: out STD_LOGIC;
10     COUT: out STD_LOGIC
11   );
12 end Practica_5;
13
14 architecture Behavioral of Practica_5 is
15
16 begin
17
18   SUM <= (A XOR B) XOR C;
19   COUT <= ((A XOR B) AND C) OR (A AND B);
20
21 end Behavioral;

```

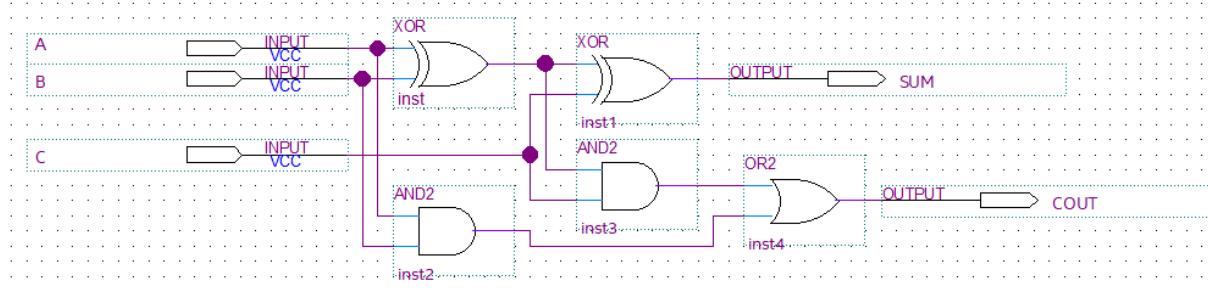
- Diagrama RTL del código VHDL



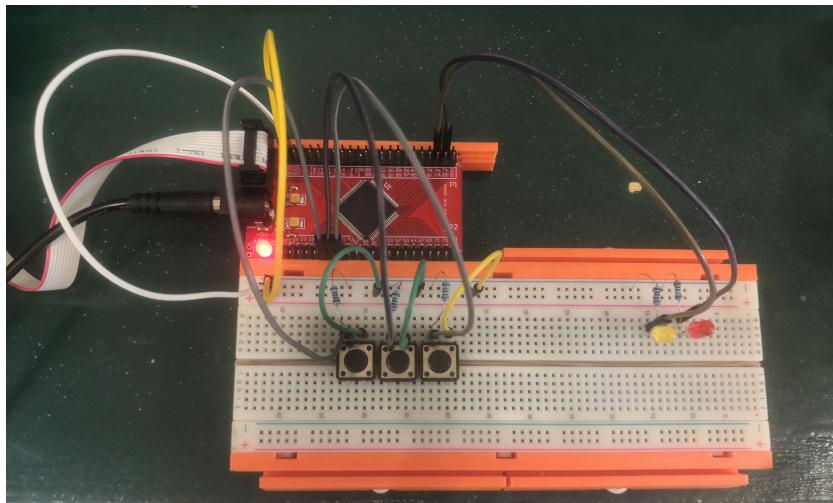
- Configuración de los pines 64 (Entrada) y 15 (Salida)



- Configuración del circuito mediante diagrama



- Armado y funcionamiento físico del circuito: <https://youtu.be/P5ZyQ9-vDBM>



NOTA: Es importante aclarar que durante esta práctica, el CPLD presentó fallas que aún no logramos solucionar, por lo que no fue posible mostrar y grabar el circuito en funcionamiento, sin embargo se incluye y muestran imágenes del circuito físico así como la implementación del mismo a través de Arduino para mostrar lo que debería hacer el circuito.

CONCLUSIÓN

Mediante esta práctica pudimos tener una mejor comprensión acerca del funcionamiento de un sumador completo de un bit. Además, nos permitió poner en práctica las dos formas de trabajar con el software para configurar el comportamiento del circuito, mediante código VHDL o a través de diagrama de circuito lógicos. El trabajar con el CPLD facilita mucho la tarea de configurar un sumador de un bit, pues las operaciones lógicas ocurren en su interior en base al comportamiento que nosotros indicamos con el código o el diagrama, lo que ahorra muchas conexiones extra que se hubiesen tenido que realizar si trabajamos sin CPLD, solo con compuertas lógicas. En nuestro caso particular, a pesar de realizar correctamente los procedimientos, tanto del código como del diagrama, no logramos ver el comportamiento en el circuito físico debido a que no logramos que el software detectara el blaster para cargar el archivo a pesar de probar en computadoras distintas, por lo que sospechamos de una posible falla en la tablilla.

BIBLIOGRAFÍA

Documentación de práctica:

[https://aulas3.uach.mx/pluginfile.php/1116743/mod_assign/introattachment/0/
Practica%20%235%20-%20Sumador.pdf?forcedownload=1](https://aulas3.uach.mx/pluginfile.php/1116743/mod_assign/introattachment/0/Practica%20%235%20-%20Sumador.pdf?forcedownload=1)